HAL
open science

Ontology matching: state of the art and future challenges

Pavel Shvaiko, Jérôme Euzenat

To cite this version:

Pavel Shvaiko, Jérôme Euzenat. Ontology matching: state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, 2013, 25 (1), pp.158-176. 10.1109/TKDE.2011.253 . hal-00917910

HAL Id: hal-00917910

https://inria.hal.science/hal-00917910

Submitted on 12 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ontology matching: state of the art and future challenges

Pavel Shvaiko and Jérôme Euzenat

Abstract—After years of research on ontology matching, it is reasonable to consider several questions: *is the field of ontology matching still making progress? Is this progress significant enough to pursue further research?* If so, *what are the particularly promising directions?* To answer these questions, we review the state of the art of ontology matching and analyze the results of recent ontology matching evaluations. These results show a measurable improvement in the field, the speed of which is albeit slowing down. We conjecture that significant improvements can be obtained only by addressing important challenges for ontology matching. We present such challenges with insights on how to approach them, thereby aiming to direct research into the most promising tracks and to facilitate the progress of the field.

Index Terms—Semantic heterogeneity, semantic technologies, ontology matching, ontology alignment, schema matching.

1 Introduction

The progress of information and communication technologies has made available a huge amount of disparate information. The problem of managing heterogeneity among various information resources is increasing. For example, most of the database research self-assessment reports recognize that the thorny question of *semantic heterogeneity*, that is of handling variations in meaning or ambiguity in entity interpretation, remains open [1]. As a consequence, various solutions have been proposed to facilitate dealing with this situation, and specifically, to automate integration of distributed information sources. Among these, semantic technologies have attracted particular attention. In this paper we focus on a kind of semantic technologies, namely, ontology matching.

An *ontology* typically provides a vocabulary that describes a domain of interest and a specification of the meaning of terms used in the vocabulary. Depending on the precision of this specification, the notion of ontology encompasses several data and conceptual models, including, sets of terms, classifications, thesauri, database schemas, or fully axiomatized theories [2]. When several competing ontologies are used in different applications, most often these applications cannot immediately interoperate. In this paper we consider ontologies expressed in OWL as a typical example of a knowledge representation language on which most of the issues can be illustrated. OWL is succeeding to a large degree as a knowledge representation standard, for instance, used for building knowledge systems. However, several matching systems discussed in the paper are able to deal with RDFS or SKOS as well. Database schemas and ontologies share similarity since they both provide a vocabulary of terms and somewhat constrain the meaning of terms used in the vocabulary. Hence, they often share similar matching solutions [3–7]. Therefore, we discuss in this paper approaches that come from semantic web and artificial intelligence as well as from databases.

Overcoming semantic heterogeneity is typically achieved in two steps, namely: (i) matching entities to determine an alignment, i.e., a set of correspondences, and (ii) interpreting an alignment according to application needs, such as data translation or query answering. We focus only on the matching step.

Ontology matching is a solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, or data translation. Thus, matching ontologies enables the knowledge and data expressed with respect to the matched ontologies to interoperate [2]. Diverse solutions for matching have been proposed in the last decades [8, 9]. Several recent surveys [10–16] and books [2, 7] have been written on the topic[1] as well.

As evaluations of the recent years indicate, the field of ontology matching has made a measurable improvement, the speed of which is albeit slowing down. In order to achieve similar or better results in the forthcoming years, actions have to be taken. We believe this can be done through addressing specifically promising challenges that we identify as: (i) large-scale matching evaluation, (ii) efficiency of matching techniques, (iii) matching with background

- *Pavel Shvaiko is with TasLab, Informatica Trentina SpA. Via G. Gilli 2, 38121 Trento, Italy. E-mail: pavel.shvaiko@infotn.it*
- *Jérôme Euzenat is with INRIA & LIG. 655 avenue de l'Europe, 38334 Saint-Ismier, France. Email: jerome.euzenat@inria.fr*

1. See http://www.ontologymatching.org for more details on the topic.

knowledge, (iv) matcher selection, combination and tuning, (v) user involvement, (vi) explanation of matching results, (vii) social and collaborative matching, $(viii)$ alignment management: infrastructure and support.

This article is an expanded and updated version of an earlier invited conference paper [17]. The first contribution of this work is a review of the state of the art backed up with analytical and experimental comparisons. Its second contribution is an in-depth discussion of the challenges in the field, of the recent advances made in the areas of each of the challenges, and an outline of potentially useful approaches to tackle the challenges identified.

The remainder of the paper is organized as follows. Section 2 presents the basics of ontology matching. Section 3 outlines some ontology matching applications. Sections 4 and 5 discuss the state of the art in ontology matching together with analytical and experimental comparisons. Section 6 overviews the challenges of the field, while Sections 7–14 discuss them in detail. Finally, Section 15 provides the major conclusions.

2 The Ontology Matching Problem

In this section we first discuss a motivating example (§2.1) and then we provide some basics of ontology matching (§2.2).

2.1 Motivating example

In order to illustrate the matching problem let us use the two simple ontologies, $O1$ and $O2$, of Figure 1. Classes are shown in rectangles with rounded corners, e.g., in $O1$, Book being a specialization (subclass) of Product, while relations are shown without the latter, such as price being an attribute defined on the integer domain and creator being a property. Albert Camus: La chute is a shared instance. Correspondences are shown as thick arrows that link an entity from $O1$ with an entity from $O2$. They are annotated with the relation that is expressed by the correspondence: for example, Person in $O1$ is less general ($\sqsubseteq$) than Human in $O2$.

Assume that an e-commerce company acquires another one. Technically, this acquisition requires the integration of their information sources, and hence, of the ontologies of these companies. The documents or instance data of both companies are stored according to ontologies $O1$ and $O2$, respectively. In our example these ontologies contain subsumption statements, property specifications and instance descriptions. The first step in integrating ontologies is matching, which identifies correspondences, namely the candidate entities to be merged or to have subsumption relationships under an integrated ontology. Once the correspondences between two ontologies have been determined, they may be used, for instance, for generating query expressions that automatically translate instances of these ontologies under an integrated ontology [18]. For example, the attributes with labels title in $O1$ and in $O2$ are the candidates to be merged, while the class with label Monograph in $O2$ should be subsumed by the class Product in $O1$.

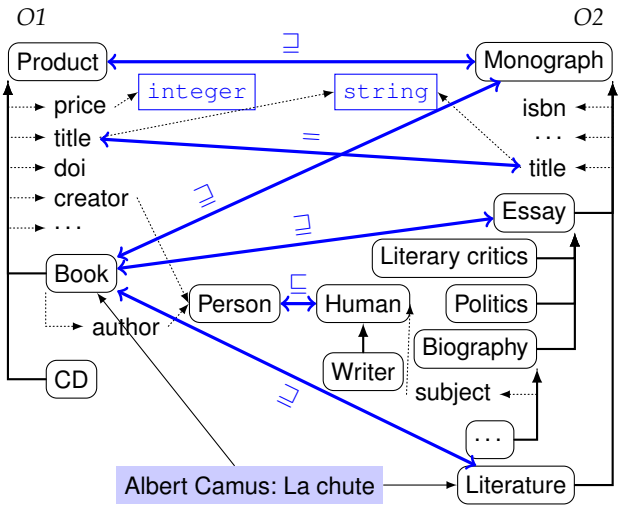

Fig. 1: Two simple ontologies and an alignment.

2.2 Problem statement

There have been different formalizations of the matching operation and its result [11, 14, 19–21]. We follow the work in [2] that provided a unified account over the previous works.

The *matching* operation determines an *alignment* A' for a pair of ontologies $O1$ and $O2$. Hence, given a pair of ontologies (which can be very simple and contain one entity each), the *matching task* is that of finding an alignment between these ontologies. There are some other parameters that can extend the definition of matching, namely: (i) the use of an input alignment A, which is to be extended; (ii) the matching parameters, for instance, weights, or thresholds; and (iii) external resources, such as common knowledge and domain specific thesauri, see Figure 2.

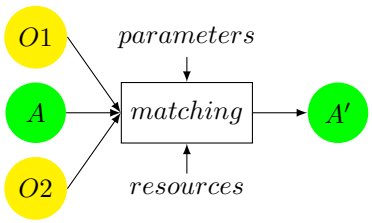

Fig. 2: The ontology matching operation.

We use interchangeably the terms matching operation, thereby focussing on the input and the result; matching task, thereby focussing on the goal and the insertion of the task in a wider context; and matching process, thereby focussing on its internals.

It can be useful to specifically consider matching more than two ontologies within the same process [22], though this is out of the scope of this paper.

An *alignment* is a set of correspondences between entities belonging to the matched ontologies. Alignments can be of various cardinalities: 1:1 (one-to-one), 1:m (one-to-many), n:1 (many-to-one) or n:m (many-to-many).

Given two ontologies, a *correspondence* is a 4-uple:

$$\langle id, e_1, e_2, r \rangle,$$

such that:

- id is an identifier for the given correspondence;
- e_1 and e_2 are entities, e.g., classes and properties of the first and the second ontology, respectively;
- r is a relation, e.g., equivalence ($=$), more general ($\sqsupseteq$), disjointness ($\perp$), holding between e_1 and e_2.

The correspondence $\langle id, e_1, e_2, r \rangle$ asserts that the relation r holds between the ontology entities e_1 and e_2. For example, $\langle id_{7,1}, \text{Book}, \text{Monograph}, \sqsupseteq \rangle$ asserts that Book in $O1$ is more general ($\sqsupseteq$) than Monograph in $O2$. Correspondences have some associated *metadata*, such as the correspondence author name. A frequently used metadata element is a confidence in the correspondence (typically in the $[0, 1]$ range). The higher the confidence, the higher the likelihood that the relation holds.

3 Applications

Ontology matching is an important operation in traditional applications, e.g., ontology evolution [23], ontology integration [24], data integration [25], and data warehouses [26]. These applications are characterized by heterogeneous models, e.g., database schemas or ontologies, that are analyzed and matched manually or semi-automatically at *design time*. In such applications, matching is a prerequisite to running the actual system.

There are some emerging applications that can be characterized by their dynamics, such as peer-to-peer information sharing [27], web service composition [28], search [29], and query answering [22]. Such applications, contrary to traditional ones, require (ultimately) a *run time* matching operation and take advantage of more explicit conceptual models. A detailed description of these applications as well as of the requirements they pose to matching can be found in [2]. We illustrate only some of these applications with the help of two short real-world examples in order to facilitate the comprehension of the forthcoming material.

Cultural heritage. A typical situation consists of having several large thesauri, such as: *Iconclass*[2] (25.000 entities) and the *Aria* collection (600 terms) from the Rijksmuseum[3]. The documents indexed by these thesauri are illuminated manuscripts and masterpieces, i.e., image data. The labels are gloss-like, i.e., sentences or phrases describing the concept, since they have to capture what is depicted on a masterpiece. Examples of labels from *Iconclass* include: city-view, and landscape with man-made constructions and earth, world as celestial body. In contrast to *Iconclass*, *Aria* uses simple terms as labels. Examples of these include: landscapes, personifications and wild animals. Matching between these thesauri (that can be performed at design time) is required in order to enable an integrated access to the masterpieces of both collections. Specifically, alignments can be used as navigation links within a multi-faceted browser to access a collection via thesauri it was not originally indexed with [30].

Geo-information (GI). A typical situation at a urban planning department of a public administration consists of a simple keyword-like request for a map generation, such as: "hydrography, Trento, January 2011". This request is a set of terms covering spatial (Trento) and temporal (January 2011) aspects to be addressed while looking for a specific theme, that is of hydrography. Handling such a request involves interpreting at run time the user query and creating an alignment between the relevant GI resources, such as those having up to date (January 2011) topography and hydrography maps of Trento in order to ultimately compose these into a single one. Technically, alignments are used in such a setting for query expansion. For what concerns thematic part, e.g., hydrography, standard matching technology can be widely reused [2, 32–34], while the spatial and temporal counterparts that constitute the specificity of GI applications have not received enough attention so far in the ontology matching field (with exceptions, such as [35, 36]), and hence, this gap will have to be covered in future.

4 Recent Matching Systems

We now review several state of the art matching systems (§4.1–§4.7) that appeared in the recent years and have not been covered by the previous surveys (§1).

Among the several dozens of systems that have appeared in these recent years, we selected some which (i) have repeatedly participated to the Ontology Alignment Evaluation Initiative (OAEI) campaigns[4] (see §5) in order to have a basis for comparisons and (ii) have corresponding archival publications, hence the complete account of these works is also available.

An overview of the considered systems is presented in Table 1. The first half of the table provides a general outlook over the systems. The *input* column presents the input format used by the systems, the *output* column describes the cardinality of the computed alignment (see §2.2), the *GUI* column shows if a system is equipped with a graphical user interface,

2. http://www.iconclass.nl/
3. http://www.rijksmuseum.nl/collectie/index.jsp?lang=en
4. http://oaei.ontologymatching.org

System	Input	Output	GUI	Operation	Terminological	Structural	Extensional	Semantic
SAMBO §4.1	OWL	1:1 alignments	Yes	Ontology merging	n-gram, edit distance, UMLS, WordNet	Iterative structural similarity based on *is-a*, *part-of* hierarchies	Naive Bayes over documents	-
Falcon §4.2	RDFS, OWL	1:1 alignments	-	-	I-SUB, Virtual documents	Structural proximities, clustering, GMO	Object similarity	-
DSsim §4.3	OWL, SKOS	1:1 alignments	AQUA Q/A [31]	Question answering	Tokenization, Monger-Elkan, Jaccard, WordNet	Graph similarity based on leaves	-	Rule-based fuzzy inference
RiMOM §4.4	OWL	1:1 alignments	-	-	Edit distance, vector distance, WordNet	Similarity propagation	Vector distance	-
ASMOV §4.5	OWL	n:m alignments	-	-	Tokenization, string equality, Levenstein distance, WordNet, UMLS	Iterative fix point computation, hierarchical, restriction similarities	Object similarity	Rule-based inference
Anchor-Flood §4.6	RDFS, OWL	1:1 alignments	-	-	Tokenization, string equality, Winkler-based sim., WordNet	Internal, external similarities; iterative anchor-based similarity propagation	-	-
AgreementMaker §4.7	XML, RDFS, OWL, N3	n:m alignments	Yes	-	TF·IDF, edit distance, substrings, WordNet	Descendant, sibling similarities	-	-

TABLE 1: Analytical comparison of the recent matching systems.

and the *operation* column describes the ways in which a system can process alignments. The second half of the table classifies the available matching methods depending on which kind of data the algorithms work on: strings (*terminological*), structure (*structural*), data instances (*extensional*) or models (*semantics*). Strings and structures are found in the ontology descriptions, e.g., labels, comments, attributes and their types, relations of entities with other entities. Instances constitutes the actual population of an ontology. Models are the result of semantic interpretation and usually use logic reasoning to deduce correspondences. Table 1 illustrates particular matching methods employed by the systems under consideration. Below, we discuss these systems in more details.

4.1 SAMBO (Linköpings U.)

SAMBO is a system for matching and merging biomedical ontologies [37]. It handles ontologies in OWL and outputs 1:1 alignments between concepts and relations. The system uses various similarity-based matchers, including:

- *terminological*: n-gram, edit distance, comparison of the lists of words of which the terms are composed. The results of these matchers are combined via a weighted sum with pre-defined weights;
- *structural*, through an iterative algorithm that checks if two concepts occur in similar positions with respect to *is-a* or *part-of* hierarchies relative to already matched concepts, with the intuition that the concepts under consideration are likely to be similar as well;
- *background knowledge based*, using (i) a relationship between the matched entities in UMLS (Unified Medical Language System) [38] and (ii) a corpus of knowledge collected from the published literature exploited through a naive Bayes classifier.

The results produced by these matchers are combined based on user-defined weights. Then, filtering based on thresholds is applied to come up with an alignment suggestion, which is further displayed to the user for feedback (approval, rejection or modification). Once matching has been accomplished, the system can merge the matched ontologies, compute the consequences, check the newly created ontology for consistency, etc. SAMBO has been subsequently extended into a toolkit for evaluation of ontology matching strategies, called KitAMO [39].

4.2 Falcon (Southeast U.)

Falcon is an automatic divide-and-conquer approach to ontology matching [40]. It handles ontologies in RDFS and OWL. It has been designed with the goal of dealing with large ontologies (of thousands of entities). The approach operates in three phases: (i) partitioning ontologies, (ii) matching blocks, and (iii) discovering alignments. The first phase starts with a structure-based partitioning to separate entities (classes and properties) of each ontology into a set of small clusters. Partitioning is based on structural proximities between classes and properties, e.g., how closely are the classes in the hierarchies of rdfs:subClassOf relations and on an extension of the Rock agglomerative clustering algorithm [41]. Then it constructs blocks out of these clusters. In the second phase the blocks from distinct ontologies are matched based on anchors (pairs of entities matched in advance), i.e., the more anchors are found between two blocks, the more similar the blocks are. In turn, the anchors are discovered by matching entities with the help of the I-SUB string comparison technique [42].

The block pairs with high similarities are selected based on a cutoff threshold. Notice that each block is just a small fragment of an ontology. Finally, in the third phase the results of the so-called V-Doc (a linguistic matcher) and GMO (an iterative structural matcher) techniques are combined via sequential composition to discover alignments between the matched block pairs. Ultimately, the output alignment is extracted through a greedy selection.

4.3 DSSim (Open U., Poznan U. of Economics)

DSSim is an agent-based ontology matching framework. The system handles large-scale ontologies in OWL and SKOS and computes 1:1 alignments with equivalence and subsumption relations between concepts and properties. It uses the Dempster-Shafer [43] theory in the context of query answering [44, 45]. Specifically, each agent builds a belief for the correctness of a particular correspondence hypothesis. Then, these beliefs are combined into a single more coherent view in order to improve correspondence quality. The ontologies are initially partitioned into fragments. Each concept or property of a first ontology fragment is viewed as a query, which is expanded based on hypernyms from WordNet [46], viewed as background knowledge. These hypernyms are used as variables in the hypothesis to enhance the beliefs. The expanded concepts and properties are matched syntactically to the similar concepts and properties of the second ontology in order to identify a relevant graph fragment of the second ontology. Then, the query graph of the first ontology is matched against the relevant graph fragment of the second ontology. For that purpose, various terminological similarity measures are used, such as Monger-Elkan and Jaccard distances, which are combined using Dempster's rule. Similarities are viewed as different experts in the evidence theory and are used to assess quantitative similarity values (converted into belief mass functions) that populate the similarity matrices. The resulting correspondences are selected based on the highest belief function over the combined evidences. Eventual conflicts among beliefs are resolved by using a fuzzy voting approach equipped with four ad hoc *if-then* rules. The system does not have a dedicated user interface but uses that of the AQUA question answering system [31] able to handle natural language queries.

4.4 RiMOM (Tsinghua U., Hong Kong U. of Science and Technology)

RiMOM is a dynamic multi-strategy ontology matching framework [47]. It extends a previous version of the system [48] that focused on combining multiple matching strategies, through risk minimization of Bayesian decision. The new version [47] quantitatively estimates the similarity characteristics for each matching task. These characteristics are used for dynamicly selecting and combining the multiple matching methods. Two basic matching methods are employed: (i) linguistic similarity (edit distance over entity labels, vector distance among comments and instances of entities) and (ii) structural similarity (a variation of Similarity Flooding [49] implemented as three similarity propagation strategies: concept-to-concept, property-to-property and concept-to-property). In turn, the strategy selection uses label and structure similarity factors, obtained as a preprocessing of the ontologies to be matched, in order to determine what information should be employed in the matching process. Specifically, the strategy selection dynamically regulates the concrete feature selection for linguistic matching, the combination of weights for similarity combination, and the choice of the concrete similarity propagation strategy. After similarity propagation, the matching process concludes with alignment refinement and extraction of the final result.

4.5 ASMOV (INFOTECH Soft, Inc., U. of Miami)

ASMOV (Automatic Semantic Matching of Ontologies with Verification) is an automatic approach for ontology matching that targets information integration for bioinformatics [50]. Overall, the approach can be summarized in two steps: (i) similarity calculation, and (ii) semantic verification. It takes as input two OWL ontologies and an optional input alignment and returns as output an n:m alignment between ontology entities (classes and properties). In the first step it uses lexical (string equality, a variation of Levenshtein distance), structural (weighted sum of the domain and range similarities) and extensional matchers to iteratively compute similarity measures between two ontologies, which are then aggregated into a single one as a weighted average. It also uses several sources of general and domain specific background knowledge, such as WordNet and UMLS, to provide more evidence for similarity computation. Then, it derives an alignment and checks it for inconsistency. Consistency checking is pattern based, i.e., that instead of using a complete solver, the system recognizes sets of correspondences that are proved to lead to an inconsistency. The semantic verification process examines five types of patterns, e.g., disjoint-subsumption contradiction, subsumption incompleteness. This matching process is repeated with the obtained alignment as input until no new correspondences are found.

4.6 Anchor-Flood (Toyohashi U. of Technology)

The Anchor-Flood approach aims at handling efficiently particularly large ontologies [51]. It inputs ontologies in RDFS and OWL and outputs 1:1 alignments. The system starts with a pair of similar concepts from two ontologies called an anchor, e.g., all exactly matched normalized concepts are considered as anchors. Then, it gradually proceeds by analyzing the

neighbors, i.e., super-concepts, sub-concepts, siblings, of each anchor, thereby building small segments (fragments) out of the ontologies to be matched. The size of the segments is defined dynamically starting from an anchor and exploring the neighboring concepts until either all the collected concepts are explored or no new matching pairs are found. The system focuses on (local) segment-to-segment comparisons, thus it does not consider the entire ontologies which improves the system scalability. It outputs a set of correspondences between concepts and properties of the semantically connected segments. For determining the correspondences between segments the approach relies on terminological (WordNet and Winkler-based string metrics) and structural similarity measures, which are further aggregated by also considering probable misalignments. The similarity between two concepts is determined by the ratio of the number of terminologically similar direct super-concepts on the number of total direct super-concepts. Retrieved (local) matching pairs are considered as anchors for further processing. The process is repeated until there are no more matching pairs to be processed.

4.7 AgreementMaker (U. of Illinois at Chicago)

AgreementMaker is a system comprising a wide range of automatic matchers, an extensible and modular architecture, a multi-purpose user interface, a set of evaluation strategies, and various manual, e.g., visual comparison, and semi-automatic features, e.g., user feedback [52]. It has been designed to handle large-scale ontologies based on the requirements coming from various domains, such as the geospatial and biomedical domains. The system handles ontologies in XML, RDFS, OWL, N3 and outputs 1:1, 1:m, n:1, n:m alignments. In general, the matching process is organized into two modules: similarity computation and alignment selection. The system combines matchers using three layers:

- The matchers of the first layer compare concept features, such as labels, comments, instances, which are represented as TF·IDF vectors used with a cosine similarity metric. Other string-based measures, e.g., edit distance, substrings, may be used as well.
- The second layer uses structural ontology properties and includes two matchers called descendants similarity inheritance (if two nodes are matched with high similarity, then the similarity between the descendants of those nodes should increase) and siblings similarity contribution (which uses the relationships between sibling concepts) [33].
- At the third layer, a linear weighted combination is computed over the results coming from the first two layers, whose results are further pruned based on thresholds and desired output cardinalities of the correspondences.

The system has a sophisticated user interface deeply integrated with the evaluation of ontology alignment quality, being an integral part of the matching process, thus empowering users with more control over it.

4.8 Analytical summary

The following can be observed concerning the considered systems (§4.1–§4.7, see also Table 1):

- The approaches equally pursue the development of generic matchers, e.g., Falcon, RiMOM, Anchor-Flood, as well as those focusing on particular application domains, e.g., SAMBO, ASMOV, that target primarily biomedical applications.
- Most of the systems under consideration declare to be able to handle efficiently large-scale ontologies, i.e., tens of thousands of entities (see some experimental comparisons in §5). This is often achieved through employing various ontology partitioning and anchor-based strategies, such as in Falcon, DSSim or Anchor-Flood.
- Although all systems can deal with OWL (being an OAEI requirement), many of them can be applied to RDFS or SKOS.
- Most of the systems focus on discovering 1:1 alignments, but yet several systems are able to discover n:m alignments. Moreover, most of the systems focus on computing equivalence relations, with the exception of DSSim, which is also able to compute subsumption relations.
- Many systems are not equipped with a graphical user interface, with several exceptions, such as SAMBO, DSSim, and AgreementMaker.
- Semantic and extensional methods are still rarely employed by the matching systems. In fact, most of the approaches are quite often based only on terminological and structural methods.
- Many systems have focussed on combining and extending the known methods. For example, the most popular of these are variations of edit distance and WordNet matchers as well as iterative similarity propagation as adaptation of the Similarity Flooding algorithm. Thus, the focus was not on inventing fundamentally new methods, but rather on adapting and extending the existing methods.

5 Recent Matching Evaluations

We provide a comparative experimental review of the matching systems described previously (§4) in order to observe and measure empirically the progress made in the field. We base our analysis on the Ontology Alignment Evaluation Initiative (OAEI) and more precisely on its 2004–2010 campaigns [53–59]. OAEI is a coordinated international initiative that

organizes annual evaluations of the increasing number of matching systems. It proposes matching tasks to participants and their results are evaluated with measures inspired from information retrieval. These are precision (which is a measure of correctness), recall (which is a measure of completeness) and F-measure, which aggregates them.

We consider here the three oldest test cases of OAEI in order to have a substantial set of data for comparison as well as diversity in tasks from automatically generated test cases to expressive ontologies. These are: benchmarks (§5.1), web directories (§5.2) and anatomy (§5.3). Participants were allowed to use one algorithm and the same set of parameters for all the test cases. Beside parameters, the input of the algorithms must be two ontologies to be matched and any general purpose resource available to everyone, i.e., resources designed especially for the test cases were not allowed, see for further details [53].

5.1 Benchmarks

The goal of the benchmark test case is to provide a stable and detailed picture of each matching algorithm. For that purpose, the algorithms are run on systematically generated test cases.

Test data. The domain of this test case is bibliographic references. It aims at comparing an OWL-DL ontology containing more than 80 entities with its variations. Most of the variations are obtained by discarding features of the original ontology. Other variations select either unrelated ontologies or other available ontologies on the same topic.

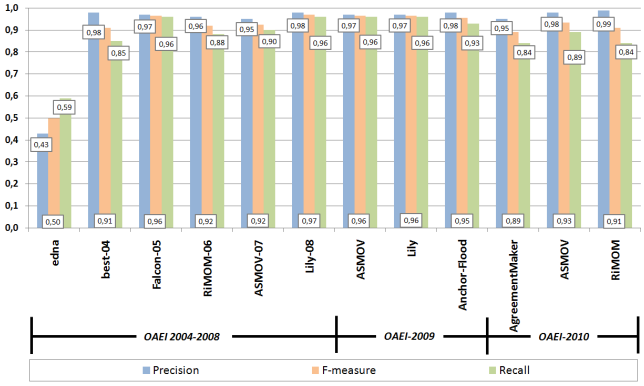

Fig. 3: Benchmarks: comparison of matching quality results in 2004–2010. More systems are mentioned in the figure with respect to those presented in §4. The results of these systems are given for the completeness of the presentation, see for further details [53, 59].

Evaluation results. A comparative summary of the best results of OAEI on the benchmarks is shown in Figure 3. *edna* is a simple edit distance algorithm on labels, which is used as a baseline. For 2004, we maximized the results of the two best systems Fujitsu and PromptDiff. The two best systems of the last several years are ASMOV [50] and Lily [60]. Their results are very comparable. A notable progress has been made between 2004 and 2005 by Falcon; and the results of 2005 were repeated in 2009 by both ASMOV and Lily.

5.2 Directory

The directory test case aims at providing a challenging task in the domain of large directories constructed from Google, Yahoo and Looksmart web directories. These directories have vague terminology and modeling principles, thus, the matching tasks incorporate the typical uncontrolled open web modeling and terminological errors. The test case was built following the TaxMe2 methodology [61].

Test data. The data set is presented as taxonomies where the nodes of the web directories are modeled as classes and the classification relation connecting the nodes is modeled as rdfs:subClassOf. There are more than 4.500 node matching tasks, where each node matching task is composed from the paths to the root of the nodes in the web directories.

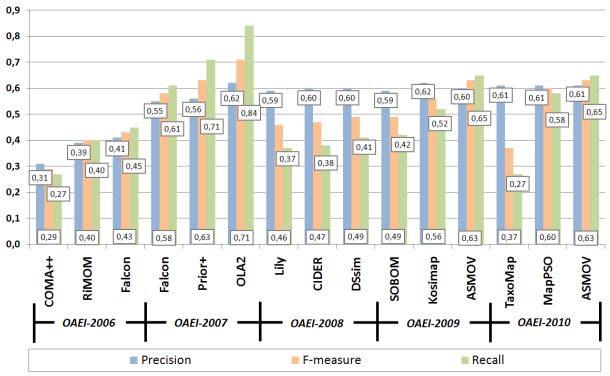

Fig. 4: Directory: comparison of matching quality results in 2006–2010. More systems are mentioned in the figure with respect to those presented in §4. The results of these systems are given for the completeness of the presentation, see for further details [53, 59].

Evaluation results. A comparison of the results in 2006–2010 for the top-3 systems of each year based on the highest F-measure is shown in Figure 4. A key observation is that from 2006 to 2007 we can measure a continuous improvement of the results, while in 2008 and 2009 the participating systems have either maintained or decreased their F-measure values. The quality of the best F-measure result of 2009 (0.63) achieved by ASMOV is higher than the best F-measure of 2008 (0.49) demonstrated by DSSim [45]. It is higher than that of 2006 by Falcon (0.43). It equals to that of Prior+ [62] and is still lower than the best F-measure of 2007 (0.71) by OLA_2 [63].

5.3 Anatomy

The focus of this test case is to confront existing matching technology with expressive ontologies in the biomedical domain. Two of its specificities are the specialized vocabulary of the domain and the usage of OWL modeling capabilities.

System	Benchmarks (§5.1) 2006	2007	2008	2009	2010	$\pm\%_B$	Directory (§5.2) 2006	2007	2008	2009	2010	$\pm\%_D$	Anatomy (§5.3) 2007	2008	2009	2010	$\pm\%_A$	Average $\pm\%$
SAMBO (§4.1)		0.71	0.88			+24							0.82	0.85			+4	+14
Falcon (§4.2)	0.89	0.89			0.73	-18	0.43	0.58				+35	0.74				n/a	+8
DSsim (§4.3)	0.70	0.77	0.92	0.91		+30		0.41	0.49	0.49		+19	0.20	0.62	0.75		+275	+108
RiMOM (§4.4)	0.92	0.91	0.96	0.96	0.91	-1	0.4	0.55	0.26			-35	0.48	0.82	0.79		+65	+10
ASMOV (§4.5)		0.92	0.96	0.96	0.93	+1		0.5	0.2	0.63	0.63	+26	0.75	0.71	0.75	0.79	+5	+11
Anchor-Flood (§4.6)			0.94	0.95		+1								0.77	0.75		-3	-1
AgreementMaker (§4.7)				0.93	0.89	-4							0.41		0.83	0.88	+115	+56

TABLE 2: The progress made by the considered systems over the recent years (2006–2010). For each year we report the F-measure indicator obtained by the systems on three test cases: benchmarks, directory and anatomy. The empty cells mean that the corresponding systems did not participate on a test case in a particular year. The $\pm\%$ column stands for the progress/regress made over the years, calculated as a percentage increase between the first and the last participation, e.g., for SAMBO on benchmarks resulting in $0.71 + 24\% \approx 0.88$. The last column shows the average progress made by the systems over different years on different test cases, calculated as the average over $\%_B, \%_D, \%_A$, e.g., for AgreementMaker this results in $(-4 + 115)/2 \approx +56\%$.

Test data. The ontologies are part of the Open Biomedical Ontologies (OBO) designed from the NCI Thesaurus (3304 classes) describing the human anatomy, published by the National Cancer Institute and the Adult Mouse Anatomical Dictionary (2744 classes). This test case has been used since 2007, while in 2005 and 2006 it was run on a different test data, which we do not consider here and focus on the more recent results instead.

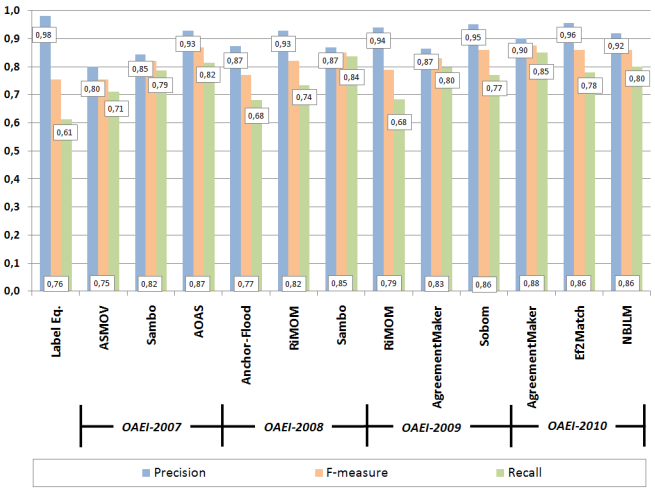

Fig. 5: Anatomy: comparison of matching quality results in 2007–2010. More systems are mentioned in the figure with respect to those presented in §4. The results of these systems are given for the completeness of the presentation, see for further details [53, 59].

Evaluation results. A comparison of the results in 2007–2010 for the top-3 systems of each year based on the highest F-measure is shown in Figure 5.

We can make two key observations. The first one is that a baseline label matcher based on string equality already provides quite good results with F-measure of 0.76. The second one is that in all the years the best F-measure remained stable around of 0.86. However, some progress have been made in terms of efficiency, i.e., the run time reduced from days and hours to minutes and seconds. For example, the best runtime result of 15s in 2009 belongs to Anchor-Flood (its F-measure was 0.75). While in 2007 and 2008 the competition was clearly dominated by the AOAS [64] and SAMBO systems that were heavily exploiting background knowledge (UMLS); in turn, in 2009 the best result showed by Sobom [65] was obtained without using any background knowledge. Finally, in 2010 the best result was shown by AgreementMaker.

5.4 Experimental summary

As we can see from the previous subsections (§5.1–§5.3) various sets of systems participate on various test cases, but not necessarily on all of these. Not all the systems participated every year, which prohibits measuring comprehensively the progress of each system over the years. In Table 2, when available, we report the F-measure obtained by these systems and the respective progress or regress made. From Table 2 we conclude that:

- Individual systems improve over the years on the same test cases. An exception includes RiMOM on the directory test case, what can be explained by the new release of the system, which still required tuning (see [47]).
- Better matching quality on one task is not achieved at the expense of another task on the average.
- The overall average improvements made by the individual systems on the test cases under considerations reach 108% increase or 28 percentage points (by DSSim) in the recent years.

An average progress over the OAEI participants that have been made in the three test cases considered from the early years to the recent years is of ∼30% in terms of F-measure (the average of all progression reported in Table 2), i.e., an increase of ∼10 percentage points on F-measure. Moreover, on the anatomy test case, the runtime improved 37 times on average from 692mn (about 11 hours) in 2007 to 18mn in 2009; see [53] for an in-depth discussion. Thus, measurable progress is observed in terms of effectiveness and efficiency made by the automatic ontology matching systems in the recent years.

At present, in the database community, there are no well-established benchmarks for comparing schema matching tools. However, there are many recent database schema matching tools and more generally model management infrastructures, e.g., COMA++ [4], AgreementMaker [52], GeRoMe [66], Harmony [67], that are able also to process ontologies, and hence, might be interested to test them within OAEI, as actually already happens, though modestly. On the other hand, OAEI has to consider including database schema matching tasks involving XML and relational schemas in order to improve the cross-fertilization between these communities.

6 TOWARDS THE CHALLENGES

After years of work on ontology matching, several questions arise: *is the field still making progress? Is this progress significant enough to pursue further research?* If so, *what are the particularly promising directions?*

The previous section showed that the field is indeed making measurable progress, but the speed of this progress is slowing down and becomes harder to determine. Also, the need for matching has risen in many different fields which have diverging demands, for instance, design time matching with correct and complete alignments, e.g., required when two banks merge, vs. run time matching with approximate alignments, e.g., acceptable in query answering on the web [2]. This calls for more precise and more versatile evaluations of matchers.

The second question interrogates the significance of the obtained results. This question requires measurements as well: the OAEI evaluations measured the progress of ∼10 percentage points in the recent five-six years (§5.4). This is a sufficient result (compared to other fields of computer science) to peruse further research into it. Also, we can see from the benchmarks, that after a few years, the improvement rate is slowing down. Hence, in order to support a similar or a stronger growth in the forthcoming years some specific actions have to be taken. In particular, we propose to guide the evolution of the ontology matching field by addressing some specific challenges.

With respect to the third question, we offer eight challenges for ontology matching (see Table 3). The challenges under consideration are among the major ontology matching topics of the recent conferences in semantic web, artificial intelligence and databases.

If the design of matchers consists of tuning further similarity measures or issuing other combinations of matchers, it is not to be expected a revolutionary progress, but most likely only an incremental one, as §5 also suggests. Other open issues are the computation of expressive alignments, e.g., correspondences across classes and properties [47, 68], oriented alignments (with non equivalence relations) [69, 70], or cross-lingual matching [71, 72]. Such issues are gradually progressing within the ontology matching field. In the first years of OAEI, it was not possible to test such alignments because there was not enough matching systems able to produce them. Only recently, oriented matching datasets were introduced and there are more systems able to produce complex correspondences. Moreover, we consider these issues as too specific with respect to the other challenges discussed, so we did not retain them as challenges.

Breakthroughs can come from either completely different settings or classes of systems particularly adapted to specific applications. We can seek for such improvements from recovering background knowledge (§9), for example, from the linked open data cloud as it represents a large and continuously growing source of knowledge. Another source of quality gains is expected from the working environment in which matching is performed. Hence, work on involving users in matching (§11) or social and collaborative matching (§13) may provide surprising results. The challenges have been selected by focusing on pragmatic issues that should help consolidating the available work in the field, bringing tangible results in the short-medium period, thus, leaving most of the theoretical and less promising directions aside. For example, in [17] we also identified as challenges uncertainty in ontology matching [73, 74] and reasoning with alignments [75, 76]. These are challenging theoretical issues, but they have a long term impact, hence, we do not discuss them here.

Another point worth mentioning is the rise of linked data [77] and the subsequent need for data interlinking. Ontology matching can take advantage of linked data as an external source of information for ontology matching, this is fully relevant to the "matching with background knowledge" challenge. Conversely, data interlinking can benefit from ontology matching by using correspondences to focus the search for potential instance level links [78]. OAEI since 2009 reacted to this need by hosting a specific instance matching track. However, data interlinking is a more specific topic, which is out of scope of this paper.

The challenges are articulated as follows. We start with the issue of evaluating ontology matching (§7), since this theme has had a large impact on the development of matchers in recent years and it shows their practical usefulness. Then, the next three challenges (§8–10) are concerned with creating better (more effective and efficient) automatic matching technology and cover, respectively, such aspects as: efficiency of ontology matching techniques (§8), involving background knowledge (§9), matcher selection, combination and self-configuration (§10). These problems have become prevalent with the advent of applications requiring run time matchers. In turn, Sections 11–13 consider matchers and alignments in their relation with users and respectively cover how and where to involve users of the matching technology (§11), and what explanations of matching results are required (§12). Moreover, users can be considered collectively when working collaboratively on alignments (§13). This, in turn, requires an alignment infrastructure for sharing and reusing alignments (§14). Solving these problems would bring ontology matching closer to final users and more prone to fill their needs.

To understand better the most pressing issues for the different types of applications (§3), Table 3 crosses the challenges identified and two broad types of applications, i.e., those requiring design time and run time matching. Half of the challenges are largely important for both design and run time applications,

Challenges vs. applications	Design time	Run time
Large-scale evaluation (§7)	√	
Efficiency of ontology matching (§8)		√
Matching with background knowledge (§9)	√	√
Matcher selection and self-configuration (§10)	√	√
User involvement (§11)	√	√
Explanations of ontology matching (§12)	√	
Collaborative and social ontology matching (§13)		√
Alignment infrastructure (§14)	√	√

TABLE 3: Applications vs. challenges. The checkmarks indicate the *primarily* impact of the challenges under consideration on two broad types of applications.

while the other half is primarily important either to one or another type of applications, thereby showing commonalities and specificities of these. For example, efficiency of ontology matching techniques is vital for run time applications, while involving background knowledge, matcher selection and self-configuration are crucial for improving quality (precision, recall) of matching results in both design and run time applications.

Each of the challenges is articulated in three parts: definition of the challenge, overview of recent advances (that complement those discussed in §4), and discussion of potential approaches to tackle the challenge under consideration.

7 Large-scale matching evaluation

The growth of matching approaches makes the issues of their evaluation and comparison more severe. In fact, there are many issues to be addressed in order to empirically prove the matching technology mature and reliable.

The challenge. Large tests involving 10.000, 100.000, and 1.000.000 entities per ontology are to be designed and conducted. In turn, this raises the issues of a wider automation for acquisition of reference alignments, e.g., by minimizing the human effort while increasing an evaluation dataset size.

We believe that the point of large-scale evaluation is of prime importance, though there are some other issues around ontology matching evaluation to be addressed as well:

- More accurate evaluation quality measures, beside *precision* and *recall* [2], are needed. Application specific measures have to be developed in order to assess whether the result of matching, e.g., F-measure of 40 or 80%, is good enough for a particular application, such as navigation among collections of masterpieces in the cultural heritage settings (§3) or web service matching. This should help quantifying more precisely the usefulness and differences between matching systems in some hard metrics, such as development time.
- Interoperability benchmarks, testing the ability of exchanging data without loss of information between the ontology matching tools, have to be designed and conducted.
- A methodology and test cases allowing for a comparative evaluation of instance-based matching approaches have to be designed and conducted.

Recent advances. OAEI campaigns gave some preliminary evidence of the scalability characteristics of the ontology matching technology. For example, most of the test cases of OAEI dealt with thousands of matching tasks, with an exception of the very large cross-lingual resources test case of OAEI-2008 [57]. Similar observations can be made as well with respect to individual matching evaluations [79–82].

Below we summarize the recent advances along the three previous issues:

- Initial steps towards better evaluation measures have already been done by proposing semantic versions of precision and recall [83] implemented in [84] and in the Alignment API [85, 86]. In turn, an early attempt to introduce application specific measures was taken in the library test case (a variation of the cultural heritage case in §3) of OAEI-2008 [24]. This is similar to the task-based evaluation used in ontology learning [87, 88].
- Despite efforts on meta-matching systems, on composing matchers [39, 89, 90], on the Alignment API [85] and in the SEALS[5] project promoting automation of evaluations, in particular for ontology matching [91], the topic of interoperability between matching tools remains largely unaddressed.
- The theme of a comparative evaluation of instance-based matching approaches is in its infancy, some test cases that have been used in the past can be found in [92–95], while a recent approach towards a benchmark for instance matching was proposed and implemented in OAEI-2009 and OAEI-2010 [58, 59, 96].

Discussion. A plausible step towards large-scale ontology matching evaluation was taken within the very large cross-lingual resources test case of OAEI-2008. In particular, it involved matching among the following three resources: (i) WordNet which is a lexical database for English, (ii) DBPedia, which is a collection of "things", each tied to an article in the English language Wikipedia, (iii) GTAA, which is a Dutch thesaurus used by the Netherlands Institute for Sound and Vision to index TV programs. The number of entities involved from each of the resources are: 82.000, 2.180.000 and 160.000, respectively. OAEI-2009 made one step further by having specific instance matching track where the whole linked open dataset[6] was involved.

5. Semantic Evaluation at Large Scale: http://www.seals-project.eu
6. http://www.linkeddata.org

Finding a large-scale real world test case is not enough for running an evaluation. A reference alignment against which the results provided by matching systems has to be created. A typical approach is to build it manually, however, the number of possible correspondences grows quadratically with the number of entities to be compared. This often makes the manual construction of the reference correspondences demanding to the point of being infeasible for large-scale matching tasks. A semi-automatic approach to the construction of reference alignment has been proposed in [61], which can be used as a starting point.

It remains difficult to know which matcher fits best to which task or application. To this end, the notion of *hardness* [97] for matching, identifying the degree of difficulty of a particular test would be useful. This would allow for automatically generating tests with particular characteristics of required hardness. This would also allow for defining test profiles (specifying dataset characteristics and measures) for different types of applications.

8 Efficiency of Matching Techniques

Beside quality, the efficiency of matchers is of prime importance in dynamic applications, especially, when a user cannot wait too long for the system to respond or when memory is limited. Current ontology matchers are mostly design time tools which are usually not optimized for resource consumption.

The challenge. The *execution time* indicates efficiency properties of matchers. However, good execution time can be achieved by using a large amount of *main memory*, or bandwidth taking on par the other computational resources, such as CPU.

Thus, usage of main memory should also be measured or improved. Moreover, we can expect the need for matching on handheld computers or smartphones in the near future. In overall, the challenge is to come up with scalable ontology matching reference solutions.

Recent advances. As Section 4 indicates, the issue of efficiency was addressed explicitly by many recent systems. However, for instance, in the anatomy track of OAEI-2007 [56], only a few systems, such as Falcon (§4.2), took several minutes to complete this matching task, while other systems took much more time (hours and even days). In OAEI-2009, Anchor-Flood (§4.6) managed to solve it in 15 seconds. In the very large cross-lingual resources test case of OAEI-2008 only DSSim (§4.3) took part (out of 13 participants), though the input files were manually split into fragments and then the matching system was applied on the pairs of these fragments.

Discussion. Efficiency issues can be tackled through a number of strategies, including:

- *parallelization* of matching tasks, e.g., cluster computing;
- *distribution* of matching tasks over peers with available computational resources;
- *approximation* of matching results, which over time become better, e.g., more complete;
- *modularization* of ontologies, yielding smaller more targeted matching tasks;
- *optimization* of existing and empirically proved-to-be-useful matching methods.

To our knowledge the first two items above remain largely unaddressed so far, and thus, will have to be covered in future. There are tasks, such as matching very large cross-lingual resources of OAEI-2008, which the existing matching technology cannot handle automatically (the resources were too large). More computing power does not necessarily improve matching quality, but, at least at the beginning, it would accelerate the first run and the analysis of the bottlenecks. To this end, the approaches taken in the LarKC project [98] to realize the strategies mentioned previously (e.g., through divide-conquer-swap strategy, which extends the traditional approach of divide-and-conquer with an iterative procedure whose result converges towards completeness over time) should be looked for and adapted to ontology matching. The existing work mainly focused on the last three items. Below, we give insights on potential further developments of the themes of approximation, modularization and optimization.

The complexity of matching (in a pair-wise set up) is usually proportional to the size of the ontologies under consideration and the number of matching algorithms employed. A straightforward approach here is to reduce the number of pair-wise comparisons in favor of a (n incomplete) top-down strategy as implemented in QOM [99], or to avoid using computationally expensive matching methods, such as in RiMOM [47] by suppressing the structure based strategies and by applying only a simple version of the linguistic based strategies.

Another worthwhile direction to avoid exhaustive pair-wise comparisons, which appears to be particulary promising when handling large ontologies, is based on segment-based approaches, e.g., COMA++ [79] and Anchor-Flood (§4.6), thus targeting at matching only the similarly enough segments. This theme has to be further and more systematically developed. It is also worth investigating how to automatically partition large ontologies into proper segments [40]. The efficiency of the integration of various matchers can be improved by minimizing (with the help of clustering, such as in PORSCHE [100] and XClust [101]) the target search space for a source ontology entity.

Optimizations are worth performing only once the underlying basic techniques are stable. For example, in the case of S-Match [5, 102] the matching problem

was reduced to the validity problem for the propositional calculus. The basic version of S-Match used a standard satisfiability procedure of SAT4J[7]. Once it has been realized that the approach is promising (based on preliminary evaluations), the efficiency problems were tackled. Specifically, for some frequent practical cases (e.g., when the propositional formula, encoding a matching problem, appears to be Horn) satisfiability can be tested in linear time, while a standard propositional satisfiability solver would require quadratic time [5]. Finally, the LogMap approach [103] used an incomplete reasoner as well as a number of optimizations to obtain the results faster, thereby exploiting several strategies to improve efficiency.

9 Matching with Background Knowledge

One source of difficulty for matching is that ontologies are designed in a particular context, with some background knowledge, which often do not become part of the final ontology specification.

The challenge. Matching can be performed by discovering a common context or background knowledge for the ontologies and use it to extract relations between ontology entities. This context can take different forms, such as a set of resources (web pages, pictures, etc.) which have been annotated with the concepts from an ontology, which provides common anchors to the ontologies to be matched. The difficulty is a matter of balance: adding context provides new information, and hence, helps increasing recall, but this new information may also generate incorrect, matches which decreases precision.

Recent advances. Various strategies have been used to deal with the lack of background knowledge. In particular:

- Declaring missing axioms manually as a pre-match effort (Cupid [3, 6], COMA [104], a cluster-based approach proposed in [105]) or using partial input alignments (SAMBO [106]).
- Reusing previous matches (COMA++ [79]). More generally storing and sharing existing alignments can be used for composing alignments, which helps solving part of the matching problem.
- Using the web as background knowledge [107], and specifically, exploiting linked data as background knowledge [108, 109] or the work on search engine weighted approximate matching [110].
- Using domain specific corpora (of schemas and mappings) [107, 111] or schema covers [112];
- Using domain specific ontologies, e.g., in the field of anatomy [64, 107], upper-level ontologies [113, 114], or all the ontologies available on the semantic web, such as in the work on Scarlet [115].

7. http://www.sat4j.org/

In addition, the work on S-Match [116] discussed an automatic approach to deal with the lack of background knowledge in matching tasks by using semantic matching [117, 118] iteratively. On top of S-Match, the work in [119] discussed the use of UMLS, instead of WordNet, as a source of background knowledge in medical applications.

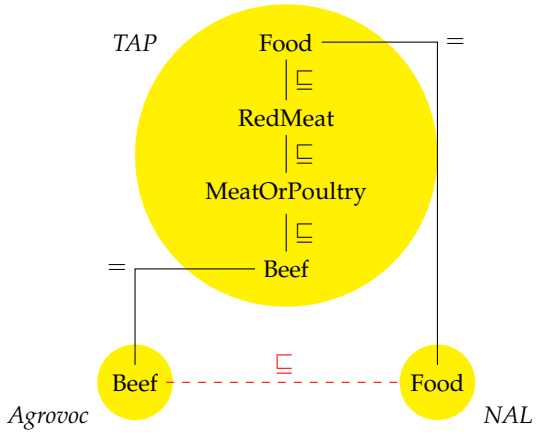

Fig. 6: Use of background knowledge in Scarlet [115]. The process is made of two steps: (i) finding an ontology referring to the concepts to be matched, (ii) inferring a relation between these concepts in function of those of the background ontology.

The techniques mentioned above have helped improving the results of matchers in various cases. For instance, Figure 6 shows two entities from the Agrovoc[8] and NAL[9] thesauri that had to be matched in the *food* test case of OAEI-2007. When considering concepts Beef and Food, the use of background knowledge found on the web, such as the TAP[10] ontology, helps deduce that Beef is less general than Food. The same result can be also obtained with the help of WordNet since Beef is a hyponym (is a kind) of Food. Thus, multiple sources of background knowledge can simultaneously help.

Discussion. The techniques mentioned before can undergo different variations based on:

- the way background knowledge sources are identified to be useful, e.g., if there are enough entities in common for a particular matching task;
- the way background knowledge sources are selected, i.e., given multiple sources, such as domain specific ontologies and upper-level ontologies, identified in the former step, selecting one or a combination of these to use;
- the way ontology entities are matched against the background knowledge sources, e.g., by employing simple string-based techniques or more sophisticated matchers;
- the way the obtained results are combined or aggregated, e.g., by majority voting.

8. http://www.fao.org/aims/ag_intro.htm
9. http://www.nal.usda.gov/
10. http://139.91.183.30:9090/RDF/VRP/Examples/tap.rdf

Once the necessary knowledge has been recovered, e.g., through a composition of several auxiliary resources, the issue is how to maintain it. Several alternatives can be explored, including: (i) extending (privately or locally) general purpose resources, such as WordNet or schema.org, towards specific domain knowledge, (ii) sharing the recovered knowledge (publicly) as linked open data.

The insights provided above have to be systematically investigated, combined in a complementary fashion and evaluated. This is particulary important in dynamic settings, where the matching input is often shallow (especially when dealing with fragmented descriptions), and therefore, incorporates fewer clues. To this end, it is vital to identify the minimal background knowledge necessary, e.g., a part of TAP in the example of Figure 6, to resolve a particular problem with sufficiently good results.

10 Matcher selection, combination and tuning

Many matchers are now available. As OAEI campaigns indicate (§5), there is no single matcher that clearly dominates others. Often these perform well in some cases and not so well in some other cases. Both for design and run time matching, it is necessary to be able to take advantage of the best configuration of matchers.

The challenge. There is evidence from OAEI (§5) that matchers do not necessarily find the same correct correspondences. Usually several competing matchers are applied to the same pair of entities in order to increase evidence towards a potential match or mismatch. This requires to solve several important problems: (i) selecting matchers and combining them, and (ii) self-configuring or tuning matchers. On top of this, for dynamic applications it is necessary to perform matcher combination and self-tuning at run time, and thus, efficiency of the configuration search strategies becomes critical. As the number of available matchers increases, the problem of their selection will become more critical, e.g., when the task will be to handle more than 50 matchers within one system.

Recent advances. The problem of matcher selection has been addressed, for example, through analytic hierarchy process [120], ad hoc rules [121, 122] or a graphical matching process editor [123]. Often the matcher selection is tackled by setting appropriate weights (in [0 1]) to matchers that are predefined in a pool (of usually at most several dozens of matchers) and to be further aggregated. So far, mostly design time toolboxes allow to do this manually [16, 79, 124].

Another approach involves ontology meta-matching [89, 125], i.e., a framework for combining a set of selected ontology matchers. Instead of least-square linear regression as in [126], the work in [81] uses a machine learning technique, called boosting (the AdaBoost algorithm) in order to select matchers from a pool to be further used in combination. Multi-agent techniques have also been used for that purpose, e.g., [127] exploits the max-sum algorithm to maximize the utility of a set of agents, while [128] uses argumentation schemes to combine matching results.

The work in [89] proposed an approach to tune a library of schema matchers at design time: given a particular matching task, it automatically tunes a matching system by choosing suitable matchers, and the best parameters to be used, such as thresholds. The work in [129] discussed consensus building after many methods have been used.

Discussion. The above mentioned problems share common characteristics: (i) the search space is very large, and (ii) the decision is made involving multiple criteria. Resolving these two problems simultaneously at run time makes ontology matching even harder.

The work on evaluation (§5) can be used in order to assess the strengths and the weaknesses of individual matchers by comparing their results with task requirements. Often, there are many different constraints and requirements applied to the matching tasks, e.g., correctness, completeness, execution time, main memory, thereby involving multi-decision criteria. The main issue is the semi-automatic combination of matchers by looking for complementarities, balancing the weaknesses and reinforcing the strengths of the components. For example, the aggregation is usually performed following a pre-defined aggregation function, such as a weighted average. Novel ways of performing aggregation with provable qualities of alignments have to be looked for in order to go beyond the incremental progress that we observed in the recent years. For example, one of the plausible directions to pursue was investigated in [130], which proposed to use a decision tree as an aggregation function, where the nodes represent the similarity measures and edges are used as conditions on the results. Such a decision tree represents a plan whose elementary operations are matching algorithms. Further issues to be addressed include investigating the automatic generation of decision trees based on an application domain.

In the web setting, it is natural that applications are constantly changing their characteristics. Therefore, approaches that attempt to tune and adapt automatically matching solutions to the settings in which an application operates are of high importance. This may involve the run time reconfiguration of a matcher by finding its most appropriate parameters, such as thresholds, weights, and coefficients. The above mentioned work in [130], also contributed to the theme of tuning. Specifically, since edges in the decision tree are used as conditions, these can be viewed as thresholds,

personalized to each matcher. Thus, various ways of encoding the matcher combination and the tuning problem have to be explored and developed further.

11 User involvement

In traditional applications, the result of matching performed at design time is screened by human users before being accepted. However, the overwhelming size of data may render this task difficult. In dynamic applications, users are generally not ontology matching specialists who can be asked to inspect the alignments. Hence, in both cases, user involvement becomes crucial.

The challenge is to design ways of involving users so that they can help the matching process without being lost in the amount of results. The issue is, both for design and run time matching, to design interaction schemes which are burdenless to the user. At design time, interaction should be both natural and complete; at run time, it should be hidden in the user task.

Recent advances. So far, there have only been few studies on how to involve users in ontology matching. The works in [131, 132] proposed to use query logs to enhance match candidate generation. Several efforts were dedicated to design time matcher interaction, such as in [79, 133]. Some recent works have focussed on the ergonomic aspect of elaborating alignments, either for designing them manually or for checking and correcting them, e.g., through learning [134, 135]. Specifically, the work in [136] proposed a graphical visualization of alignments based on cognitive studies. In turn, the work in [137] provided an environment for manually designing complex alignments through the use of connected perspective that allows for quickly deemphasizing non relevant aspects of the ontologies being matched while keeping the connections between relevant entities. The work in [138] provided the Clip tool that allows for explicitly specifying structural transformations by means of a visual language, in addition to value couplings to be associated to correspondences.

Discussion. With the development of interactive approaches the issues of their usability will become more prominent. This includes scalability of visualization [139] and better user interfaces in general, which are expected to bring higher quality gains than more accurate matching algorithms [140].

An interesting trend to follow concerning user involvement relies on final users in order to learn from them – given a matching task – what is the best system configuration to approach that task. Moreover, for dynamic applications, only the final user can help. This can be exploited by adjusting matching system parameters (§10), or by experimenting with alignment selection strategies. In order to facilitate this, matching tools have to be *configurable* and *customizable*. Hence, users themselves could improve these tools, thereby arriving to the exact solution that best fits their needs and preferences. When users are given this freedom by working on tool customization, they can also provide useful feedback to system designers. Involving final users in an active manner in a matching project would increase its *impact*, as users who recognize the actual need, also have promising ideas on how to approach it [141]. When these "lead" users want something that is not available on the market, high benefits may be expected from such endeavors.

Technically, a basic premise underlying user interaction design is that users of a matching system should be able to influence the search for an optimal alignment on various levels via unified interfaces. For example, by recommending relevant background knowledge in advance, by influencing the selection and weighting of the various matching components, by criticizing aspects of intermediate results, and by determining whether the final result is good enough to be put to use. Little attention has been devoted so far to the realization of interfaces that actually allow users to become active in these ways. Systems should be developed on the basis of continual tests with final users, and the ultimate success criterion will be the extent to which the system has value for them.

Finally, as more systems will become equipped with GUIs (see Table 1), we expect that evaluation of usability and customizability of such systems will become more prominent, e.g., included as evaluation indicators of the OAEI campaigns.

12 Explanation of matching results

In order to better edit alignments, thereby providing feedback to the system, users need to understand them. It is often not sufficient that a matcher returns an alignment, for users to understand it immediately. In order for matching systems to gain a wider acceptance and to be trusted by users, it will be necessary that they provide explanations of their results to users or to other programs that exploit them. Notice that the issues of trustworthiness and provenance become particularly important in the web settings that enable social and collaborative matching (§13).

The challenge is to provide explanations in a simple, yet clear and precise, way to the user in order to facilitate informed decision making. In particular, many sophisticated techniques used by matching systems, e.g., machine learning or discrete optimization, do not yield simple or symbolic explanations.

Recent advances. There are only a few matching systems able to provide an explanation for their results [128, 142, 143]. The solutions proposed so far focus on default explanations, explaining basic matchers, explaining the matching process, and negotiating alignments by argumentation.

More recently, the work in [80] introduced the notion of a matchability score, computed via a synthetic workload, which quantifies how well on average a given schema matches future schemas. Using the matchability score, different types of matching mistakes, such as missing a correspondence or predicting a wrong correspondence, can be analyzed, e.g., by looking into the most likely general reasons leading to them. Once the matchability score has been computed for all the entities, they are ranked by increasing scores. A *matchability report* is generated for each entity by grouping incorrect matches based on their underlying (general) reasons and by displaying (i) the reason, (ii) an example to illustrate the reason, and (iii) a suggestion of revisions to be made, thereby guiding users in revising correspondences by addressing the reported mistakes.

Finally, [144] provided a mapping design wizard that uses data examples to systematically assist integration engineers in explaining and refining alignments towards a desired specification. The key intuition behind it is that integration engineers usually understand better their data than alignments. Hence, such data examples are used in explanations (of nuances) to clarify possible variations in interpretation of alignments, by including these examples into a small number of automatically generated yes-or-no questions.

Discussion. Few works have addressed the theme of explanations in ontology matching. Therefore, the directions pursued in those works are worth considering further. In addition, an interactive environment is still needed to help users accept or revise the suggested correspondences (see §14). In this respect it would be also useful to exploit the abstraction techniques extensively and build on top of the work on explanations in recommender systems [145]. In the longer term, it would be helpful to standardize explanations of matching results in order to facilitate the interaction of matching systems with other applications.

13 Social and Collaborative Matching

In an open environment like the web, social support has been the key in solving hard and large problems. This approach can be also applied to ontology matching.

The challenge. Matching could be improved through social interaction. This may be obtained with the help of people explicitly arguing about correspondences or by implicitly voting each time a correspondence is used during an interaction. This calls for algorithms able to rank a massive amount of correspondences. The incompleteness and inconsistency of alignments will have to be dealt with in a satisfactory way. Other issues include understanding what tasks are relatively easy for humans, but difficult for machines, how to individuate and deal with malicious users, and which incentive schemes promise to facilitate user participation in establishing alignments collaboratively.

Recent advances. The work in [146] extended the notion of ontology matching to community-driven ontology matching and discussed early experiments in which a community of people can share alignments over the web, reuse them as well as argue about them by using annotations. Technically this meant, among other things, extending correspondences with *metadata* covering the correspondence author name, application domain, his or her trust value in a community, etc.

The work in [147] proposed a model (the mapping ontology) for representing correspondences (called mappings) collected from the user community and the metadata associated with it. This model was further implemented in a collaborative system in the area of bio-informatics for sharing both ontologies and correspondences. It actually brought together more than 30.000 such mappings. This allows users to share, edit, search for, and discuss the mappings. The strengths of this system are a user friendly interface with the possibility to annotate correspondences and the direct connection with the ontologies which helps users to navigate through them.

Finally, [148] proposed an application of mechanical turk to enlist the multitude of users in a community to help matching schemas by asking users simple questions, e.g., is monthly-fee-rate of type DATE?, and then learn from the answers to improve matching accuracy, e.g., in case of a positive answer to the question above by using a specific *date* matcher. On the one side, the questions should be relatively easy for users to answer and, on the other side, they must affect substantially the matching accuracy. In this vein, three types of questions were used for three different purposes: (i) to verify intermediate predictions of the system, (ii) to learn domain integrity constraints, and (iii) to verify final match candidates. Users were classified into *trusted* and *untrusted*, based on their answers to a set of (evaluation) questions with answers known in advance. The answers from trusted users were further combined using a voting scheme. Ultimately, two user participation schemes have been analyzed: a standard *volunteering* scheme and a scheme in which users have to "*pay*" by answering first several questions in order to use a desired service.

Discussion. A promising way of tackling the matching task is by taking advantage of the network effect. If it is too cumbersome for one person to come up with a correct alignment between several pairs of ontologies, this can be more easily resolved by many people together, namely: (i) each person has to do a very small amount of work, (ii) each person can

improve on what has been done by others, and (iii) errors remain in minority.

Thus, the process and the dynamics of crowdsourcing and collaborative ontology matching in the context of various applications should be studied and formalized; some steps in this direction have already been taken within the (reference alignment) consensus building sessions of the Ontology Matching workshops[11]. In general, the experiences with collaborative knowledge construction [149] and, in particular, with the collaborative development of ontologies [150, 151] as well as with the community information management systems [152] should be monitored and whenever promising adapted for collaborative matching.

The success of the social and collaborative matching techniques will largely depend on the creation of a *critical mass* of users in communities of interest that actually use them – similar to what happens to any data on the web – once an alignment has been established, either manually or automatically, it should be (publicly) shared, thereby enabling its further reuse (§14). In turn, this requires an adequate support for handling trustworthiness and provenance of alignments (§12).

14 Alignment management: infrastructure and support

Storing and sharing alignments, as well as collaborative matching, should be supported by adequate tools and infrastructure, especially in dynamic applications.

The challenge. The challenge is to provide convenient and interoperable support, on which tools and, more importantly, on which applications, can rely in order to store and share alignments. This involves using standard ways to communicate alignments and retrieve them. Hence, alignment metadata and annotations should be properly taken into account.

Recent advances. We can distinguish two types of software in alignment management: (i) the infrastructure middleware, and (ii) the support environments that provide application specific access to alignments. The support environments can be dedicated to alignment edition [133], alignment processing, alignment sharing and discussing [82, 147, 153], or model management [140]. The two levels may be kept clearly separated [154] or mixed in a single system [67, 147]. As a recent example of the latter, the Harmony workbench [67] made its way into a collaborative effort, called OpenII[12], to create a suite of open-source tools for information integration, e.g., for matching and merging.

Other systems have been designed for offering a variety of matching methods and a library of mappings [82, 133, 155]. However, these systems were meant as a component for design time integration and not as a service that can be used at run time.

In turn, the Alignment server [154] has been designed as a middleware component with these goals in mind. It supports, in particular, alignment storing, correspondence annotation and sharing. It is accessible from other tools and applications through a versatile interface (HTTP, REST, SOAP, FIPA ACL). The NeOn toolkit Alignment plug-in[13] embeds it into an environment similar to Protégé[14]. The plug-in allows to manipulate alignments locally or to access them on the Alignment server.

Finally, the Cupboard system is a run time environment for supporting networked ontology organization [156]. It allows users to find alignments on the web, to match them through the Alignment server and to register them in their own workspace. These alignments can be ranked by other users and this information is made available to all the Cupboard users.

Discussion. A first step in promoting sharing, manipulating and reusing alignments is to be able to use a standard for alignments. No such a standard exists at the moment. In a longer term, such a standard may be proposed. At present, the Alignment format [85] exploited by the OAEI evaluations is used by many systems [153]. Since it is extensible and does not make assumptions about the type of matched ontologies, e.g., it can be applied to SKOS thesauri and to any structure whose elements can be identified by URIs, it would be a natural starting point for standardization.

As soon as ontologies evolve, new alignments have to be produced following the evolution of the ontology. This can be achieved by transforming the changes made to ontologies into an alignment (from one ontology version to the next one), which can be composed with the old alignment for obtaining an updated alignment.

15 Conclusions

We have introduced the basics of ontology matching with the help of examples. We outlined various matching applications as well as ontology matching use cases developed in collaboration with final users. We discussed the state of the art in ontology matching and made some analytical and empirical comparisons. The outcome of this analysis is that ontology matching is making a measurable progress, though it is slowing down. In order to address this situation, we presented eight challenges for ontology matching, accompanied for each of these with an overview of the recent advances in the field and a discussion of the potentially useful ways to approach the challenges under consideration. We believe that addressing the outlined

11. http://om2008.ontologymatching.org/
12. http://openintegration.org/
13. http://alignapi.gforge.inria.fr/neontk/
14. http://protege.stanford.edu/

challenges should accelerate the progress of the ontology matching field. Moreover, these challenges are not isolated from each other, e.g., collaborative matching requires an alignment infrastructure, and have to be considered in relation with each other.

We expect that, as ontology matching technologies are becoming more mature, practitioners will increase their expectations and will want to experiment with them more intensively. In particular, the increasing involvement of final users in matching projects should help keeping focus on capturing the values of ontology matching based on the usefulness that it brings, and thus, accelerating progress of the field during its pre-commercial phase and ultimately turning matching technologies into mainstream.

Acknowledgements: The first author appreciates support from the Trentino as a Lab initiative of the European Network of the Living Labs at Informatica Trentina. The second author has been partially supported by the European integrated project NeOn (IST-2005-027595) and the RNTL project WebContent. We are grateful to Fausto Giunchiglia (University of Trento), Lorenzino Vaccari (European Comission, DG Environment), Mathieu d'Aquin and Marta Sabou (Open University), Chan Le Duc and Antoine Zimmermann (INRIA), Antoine Isaac, Zharko Aleksovski, Marjolein van Gendt (Vrije Universiteit Amsterdam), Malgorzata Mochol (Free University of Berlin), Juan Sequeda (University of Texas at Austin) for their insightful comments on various aspects of ontology matching covered in this paper.

REFERENCES

[1] R. Agrawal, A. Ailamaki, P. A. Bernstein, E. A. Brewer, M. J. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. J. Franklin, H. Garcia-Molina, J. Gehrke, L. Gruenwald, L. M. Haas, A. Y. Halevy, J. M. Hellerstein, Y. E. Ioannidis, H. F. Korth, D. Kossmann, S. Madden, R. Magoulas, B. C. Ooi, T. O'Reilly, R. Ramakrishnan, S. Sarawagi, M. Stonebraker, A. S. Szalay, and G. Weikum, "The Claremont report on database research," *SIGMOD Record*, vol. 37, no. 3, pp. 9–19, 2008.

[2] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer, 2007.

[3] J. Madhavan, P. Bernstein, and E. Rahm, "Generic schema matching with Cupid," in *Proc. 27th International Conference on Very Large Data Bases (VLDB)*, 2001, pp. 48–58.

[4] D. Aumüller, H.-H. Do, S. Maßmann, and E. Rahm, "Schema and ontology matching with COMA++," in *Proc. 24th International Conference on Management of Data (SIGMOD), Demo track*, 2005, pp. 906–908.

[5] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, "Semantic matching: Algorithms and implementation," *Journal on Data Semantics*, vol. IX, pp. 1–38, 2007.

[6] P. A. Bernstein, J. Madhavan, and E. Rahm, "Generic schema matching, ten years later," *PVLDB*, vol. 4, no. 11, pp. 695–701, 2011.

[7] Z. Bellahsene, A. Bonifati, and E. Rahm, Eds., *Schema Matching and Mapping*. Springer, 2011.

[8] C. Batini, M. Lenzerini, and S. Navathe, "A comparative analysis of methodologies for database schema integration," *ACM Computing Surveys*, vol. 18, no. 4, pp. 323–364, 1986.

[9] S. Spaccapietra and C. Parent, "Conflicts and correspondence assertions in interoperable databases," *SIGMOD Record*, vol. 20, no. 4, pp. 49–54, 1991.

[10] E. Rahm and P. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[11] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: the state of the art," *The Knowledge Engineering Review*, vol. 18, no. 1, pp. 1–31, 2003.

[12] N. Noy, "Semantic integration: A survey of ontology-based approaches," *SIGMOD Record*, vol. 33, no. 4, pp. 65–70, 2004.

[13] A. Doan and A. Halevy, "Semantic integration research in the database community: A brief survey," *AI Magazine*, vol. 26, no. 1, pp. 83–94, 2005, special issue on Semantic integration.

[14] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," *Journal on Data Semantics*, vol. IV, pp. 146–171, 2005.

[15] N. Choi, I.-Y. Song, and H. Han, "A survey on ontology mapping." *SIGMOD Record*, vol. 35, no. 3, pp. 34–41, 2006.

[16] A. Gal and P. Shvaiko, "Advances in ontology matching," in *Advances in Web Semantics I*, T. S. Dillon, E. Chang, R. Meersman, and K. Sycara, Eds. Springer, 2009, pp. 176–198.

[17] P. Shvaiko and J. Euzenat, "Ten challenges for ontology matching," in *Proc. 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, 2008, pp. 1163–1181.

[18] R. Fagin, L. M. Haas, M. A. Hernández, R. J. Miller, L. Popa, and Y. Velegrakis, "Clio: Schema mapping creation and data exchange," in *Conceptual Modeling: Foundations and Applications*. Springer, 2009, pp. 198–236.

[19] P. Bernstein, A. Halevy, and R. Pottinger, "A vision of management of complex models," *SIGMOD Record*, vol. 29, no. 4, pp. 55–63, 2000.

[20] M. Lenzerini, "Data integration: A theoretical perspective," in *Proc. 21st Symposium on Principles of Database Systems (PODS)*, 2002, pp. 233–246.

[21] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler, "Formalizing ontology alignment and its operations with category theory," in *Proc. 4th International Conference on Formal Ontology in Information Systems (FOIS)*, 2006, pp. 277–288.

[22] B. He and K. Chang, "Automatic complex schema matching across web query interfaces: A correlation mining approach," *Transactions on Database Systems*, vol. 31, no. 1, pp. 1–45, 2006.

[23] N. F. Noy, A. Chugh, W. Liu, and M. A. Musen, "A framework for ontology evolution in collaborative environments," in *Proc. 5th International semantic web Conference (ISWC)*, 2006, pp. 544–558.

[24] A. Isaac, S. Wang, C. Zinn, H. Matthezing, L. van der Meij, and S. Schlobach, "Evaluating thesaurus alignments for semantic interoperability in the library domain," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 76–86, 2009.

[25] P. P. Talukdar, Z. G. Ives, and F. Pereira, "Automatically incorporating new sources in keyword search-based data integration," in *Proc. of the 29th International Conference on Management of Data (SIGMOD)*, 2010, pp. 387–398.

[26] S. Dessloch, M. A. Hernández, R. Wisnesky, A. Radwan, and J. Zhou, "Orchid: Integrating schema mapping and ETL," in *Proc. 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 1307–1316.

[27] M. Atencia, J. Euzenat, G. Pirrò, and M.-C. Rousset, "Alignment-based trust for resource finding in semantic p2p networks," in *Proc. 10th International Semantic Web Conference (ISWC)*, 2011, pp. 51–66.

[28] L. Vaccari, P. Shvaiko, and M. Marchese, "A geo-service semantic integration in spatial data infrastructures," *Journal of Spatial Data Infrastructures Research*, vol. 4, pp. 24–51, 2009.

[29] Y. Kitamura, S. Segawa, M. Sasajima, S. Tarumi, and R. Mizoguchi, "Deep semantic mapping between functional taxonomies for interoperable semantic search," in *Proc. 3rd Asian Semantic Web Conference (ASWC)*, 2008, pp. 137–151.

[30] M. van Gendt, A. Isaac, L. van der Meij, and S. Schlobach, "Semantic web techniques for multiple views on heterogeneous collections: A case study," in *Proc. 10th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, 2006, pp. 426–437.

[31] V. Lopez, M. Pasin, and E. Motta, "AquaLog: An ontology-portable question answering system for the semantic web." in *Proc. 2nd European Semantic Web Conference (ESWC)*, 2005, pp. 546–562.

[32] I. F. Cruz, W. Sunna, N. Makar, and S. Bathala, "A visual tool for ontology alignment to enable geospatial interoperability," *Journal of Visual Languages and Computing*, vol. 18, no. 3, pp. 230–254, 2007.

[33] I. F. Cruz and W. Sunna, "Structural alignment methods with applications to geospatial ontologies," *Transactions in Geographic Information Science*, vol. 12, no. 6, pp. 683–711, 2008.

[34] L. Vaccari, P. Shvaiko, J. Pane, P. Besana, and M. Marchese, "An evaluation of ontology matching in geo-service applications," *GeoInformatica*, 2011.

[35] C. Parent, S. Spaccapietra, and E. Zimányi, *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer, 2006.

[36] A. Schwering, "Approaches to semantic similarity measurement between geo-spatial data - a survey," *Transactions in Geographic Information Science*, vol. 12, no. 1, pp. 5–29, 2008.

[37] P. Lambrix and H. Tan, "SAMBO – a system for aligning and merging biomedical ontologies," *Journal of Web Semantics*, vol. 4, no. 1, pp. 196–206, 2006.

[38] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, pp. 267–270, 2004.

[39] P. Lambrix and H. Tan, "A tool for evaluating ontology alignment strategies," *Journal on Data Semantics*, vol. VIII, pp. 182–202, 2007.

[40] W. Hu, Y. Qu, and G. Cheng, "Matching large ontologies: A divide-and-conquer approach," *Data and Knowledge Engineering*, vol. 67, no. 1, pp. 140–160, 2008.

[41] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," in *Proc. 15th International Conference on Data Engineering (ICDE)*, 1999, pp. 512–521.

[42] G. Stoilos, G. Stamou, and S. Kollias, "A string metric for ontology alignment," in *Proc. 4th International Semantic Web Conference (ISWC)*, 2005, pp. 624–637.

[43] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[44] M. Nagy and M. Vargas-Vera, "Towards an automatic semantic data integration: Multi-agent framework approach," in *Semantic Web*, G. Wu, Ed. In-Teh, 2010, ch. 7, pp. 107–134.

[45] M. Nagy, M. Vargas-Vera, and P. Stolarski, "Dssim results for OAEI 2009," in *Proc. 4th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2009, pp. 160–169.

[46] G. Miller, "WordNet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[47] J. Li, J. Tang, Y. Li, and Q. Luo, "Rimom: A dynamic multistrategy ontology alignment framework," *IEEE Transactoins on Knowledge and Data Engineering*, vol. 21, no. 8, pp. 1218–1232, 2009.

[48] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang, "Using Bayesian decision for ontology mapping," *Journal of Web Semantics*, vol. 4, no. 1, pp. 243–262, 2006.

[49] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: a versatile graph matching algorithm," in *Proc. 18th International Conference on Data Engineering (ICDE)*, 2002, pp. 117–128.

[50] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Journal of Web Semantics*, vol. 7, no. 3, pp. 235–251, 2009.

[51] M. S. Hanif and M. Aono, "An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size," *Journal of Web Semantics*, vol. 7, no. 4, pp. 344–356, 2009.

[52] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: Efficient matching for large real-world schemas and ontologies," *PVLDB*, vol. 2, no. 2, pp. 1586–1589, 2009.

[53] J. Euzenat, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, and C. Trojahn dos Santos, "Ontology alignment evaluation initiative: six years of experience," *Journal on Data Semantics*, vol. XV, pp. 158–192, 2011.

[54] B. Ashpole, M. Ehrig, J. Euzenat, and H. Stuckenschmidt, Eds., *Proc. K-CAP Workshop on Integrating Ontologies*, 2005.

[55] J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, O. Svab, V. Svatek, W. van Hage, and M. Yatskevich, "Results of the ontology alignment evaluation initiative 2006," in *Proc. 1st International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2006, pp. 73–95.

[56] J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W. R. van Hage, and M. Yatskevich, "Results of the ontology alignment evaluation initiative 2007," in *Proc. 2nd International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC) and Asian Semantic Web Conference (ASWC)*, 2007, pp. 96–132.

[57] C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé, C. Meilicke, J. Pane, P. Shvaiko, H. Stuckenschmidt, O. Šváb Zamazal, and V. Svátek, "Results of the ontology alignment evaluation initiative 2008," in *Proc. 3rd International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2008, pp. 73–119.

[58] J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. T. dos Santos, G. A. Vouros, and S. Wang, "Results of the ontology alignment evaluation initiative 2009," in *Proc. 4th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2009, pp. 73–126.

[59] J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Šváb Zamazal, V. Svatek, and C. Trojahn, "Results of the ontology alignment evaluation initiative 2010," in *Proc. 5th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2010, pp. 85–125.

[60] P. Wang and B. Xu, "Lily: Ontology alignment results for OAEI 2009," in *Proc. 4th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2009, pp. 186–192.

[61] F. Giunchiglia, M. Yatskevich, P. Avesani, and P. Shvaiko, "A large scale dataset for the evaluation of ontology matching systems," *The Knowledge Engineering Review*, vol. 24, no. 2, pp. 137–157, 2009.

[62] M. Mao, Y. Peng, and M. Spring, "An adaptive ontology mapping approach with neural network based constraint satisfaction," *Journal of Web Semantics*, vol. 8, no. 1, pp. 14–25, 2010.

[63] J. Euzenat and P. Valtchev, "Similarity-based ontology alignment in OWL-lite," in *Proc. 15th European Conference on Artificial Intelligence (ECAI)*, 2004, pp. 333–337.

[64] S. Zhang and O. Bodenreider, "Experience in aligning anatomical ontologies." *International Journal on Semantic Web and Information Systems*, vol. 3, no. 2, pp. 1–26, 2007.

[65] P. Xu, H. Tao, T. Zang, and Y. Wang, "Alignment results of SOBOM for OAEI 2009," in *Proc. 4th International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2009, pp. 216–223.

[66] D. Kensche, C. Quix, X. L. 0002, Y. Li, and M. Jarke, "Generic schema mappings for composition and query answering," *Data and Knowledge Engineering*, vol. 68, no. 7, pp. 599–621, 2009.

[67] P. Mork, L. Seligman, A. Rosenthal, J. Korb, and C. Wolf, "The Harmony Integration Workbench," *Journal on Data Semantics*, vol. XI, pp. 65–93, 2008.

[68] C. Ghidini and L. Serafini, "Reconciling concepts and relations in heterogeneous ontologies," in *Proc. 3rd European Semantic Web Conference (ESWC)*, 2006, pp. 50–64.

[69] P. Shvaiko, F. Giunchiglia, and M. Yatskevich, "Semantic matching with S-Match," in *Semantic Web Information Management*, R. D. Virgilio, F. Giunchiglia, and L. Tanca, Eds. Springer, 2009, pp. 183–202.

[70] V. Spiliopoulos, G. A. Vouros, and V. Karkaletsis, "On the discovery of subsumption relations for the alignment of ontologies," *Journal of Web Semantics*, vol. 8, no. 1, pp. 69–88, 2010.

[71] B. Fu, R. Brennan, and D. O'Sullivan, "Using pseudo feedback to improve cross-lingual ontology mapping," in *Proc. 8th Extended Semantic Web Conference (ESWC)*, 2011, pp. 336–351.

[72] D. Spohr, L. Hollink, and P. Cimiano, "A machine learning approach to multilingual and cross-lingual ontology matching," in *Proc. 10th International Semantic Web Conference (ISWC)*, 2011, pp. 665–680.

[73] A. Gal, *Uncertain Schema Matching*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[74] R. Tournaire, J.-M. Petit, M.-C. Rousset, and A. Termier, "Discovery of probabilistic mappings between taxonomies: Principles and experiments," *Journal on Data Semantics*, vol. XV, pp. 66–101, 2011.

[75] L. Serafini and A. Tamilin, "DRAGO: Distributed reasoning architecture for the semantic web," in *Proc. 2nd European Semantic Web Conference (ESWC)*, 2005, pp. 361–376.

[76] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, "Reasoning support for mapping revision," *Journal of Logic and Computation*, vol. 19, no. 5, pp. 807–829, 2009.

[77] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, ser. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2011.

[78] A. Nikolov, V. Uren, E. Motta, and A. de Roeck, "Overcoming schema heterogeneity between linked semantic repositories

to improve coreference resolution," in *Proc. 4th Asian Semantic Web Conference (ASWC)*, 2009, pp. 332–346.
[79] H.-H. Do and E. Rahm, "Matching large schemas: Approaches and evaluation," *Information Systems*, vol. 32, no. 6, pp. 857–885, 2007.
[80] X. Chai, M. Sayyadian, A. Doan, A. Rosenthal, and L. Seligman, "Analyzing and revising mediated schemas to improve their matchability," *PVLDB*, vol. 1, no. 1, pp. 773–784, 2008.
[81] A. Marie and A. Gal, "Boosting schema matchers," in *Proc. 16th International Conference on Cooperative Information Systems (CoopIS)*, 2008, pp. 283–300.
[82] A. Ghazvinian, N. F. Noy, C. Jonquet, N. H. Shah, and M. A. Musen, "What four million mappings can tell you about two hundred ontologies," in *Proc. 8th International Semantic Web Conference (ISWC)*, 2009, pp. 229–242.
[83] J. Euzenat, "Semantic precision and recall for ontology alignment evaluation," in *Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 248–253.
[84] D. Fleischhacker and H. Stuckenschmidt, "A practical implementation of semantic precision and recall," in *Proc. 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, 2010, pp. 986–991.
[85] J. Euzenat, "An API for ontology alignment," in *Proc. 3rd International Semantic Web Conference (ISWC)*, 2004, pp. 698–712.
[86] J. David, J. Euzenat, F. Scharffe, and C. T. dos Santos, "The alignment api 4.0," *Semantic Web Journal*, vol. 2, no. 1, pp. 3–10, 2011.
[87] R. Porzel and R. Malaka, "A task-based approach for ontology evaluation," in *Proc. of the workshop on Ontology Learning and Population at the 16th Eureopean Conference on Artificial Intelligence (ECAI)*, 2004.
[88] E. Zavitsanos, G. Paliouras, and G. A. Vouros, "Gold standard evaluation of ontology learning methods through ontology transformation and alignment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1635–1648, 2011.
[89] Y. Lee, M. Sayyadian, A. Doan, and A. Rosenthal, "eTuner: tuning schema matching software using synthetic scenarios," *The VLDB Journal*, vol. 16, no. 1, pp. 97–122, 2007.
[90] M. Ehrig, S. Staab, and Y. Sure, "Bootstrapping ontology alignment methods with APFEL," in *Proc. 4th International Semantic Web Conference (ISWC)*, 2005, pp. 186–200.
[91] J. Euzenat, C. Meilicke, H. Stuckenschmidt, and C. Trojahn dos Santos, "A web-based evaluation service for ontology matching," in *Proc. 9th International Semantic Web Conference (ISWC), Demo track*, 2010, pp. 93–96.
[92] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy, "Learning to match ontologies on the semantic web," *The VLDB Journal*, vol. 12, no. 4, pp. 303–319, 2003.
[93] J. Wang, J.-R. Wen, F. Lochovsky, and W.-Y. Ma, "Instance-based schema matching for web databases by domain-specific query probing," in *Proc. 30th International Conference on Very Large Data Bases (VLDB)*, 2004, pp. 408–419.
[94] A. Bilke and F. Naumann, "Schema matching using duplicates," in *Proc. 21st International Conference on Data Engineering (ICDE)*, 2005, pp. 69–80.
[95] H. Nottelmann and U. Straccia, "A probabilistic, logic-based framework for automated web directory alignment," in *Soft computing in ontologies and the semantic web*, ser. Studies in fuzziness and soft computing, Z. Ma, Ed. Springer, 2006, vol. 204, pp. 47–77.
[96] A. Ferrara, S. Montanelli, J. Noessner, and H. Stuckenschmidt, "Benchmarking matching applications on the semantic web," in *Proc. 8th Extended Semantic Web Conference (ESWC)*, 2011, pp. 108–122.
[97] I. Gent and T. Walsh, "Easy problems are hard," *Artificial Intelligence*, vol. 70, no. 1, pp. 335–345, 1994.
[98] E. Oren, S. Kotoulas, G. Anadiotis, R. Siebes, A. Ten Teije, and F. Van Harmelen, "Marvin: distributed reasoning over large-scale semantic web data," *Journal of Web Semantics*, vol. 7, no. 4, pp. 305–316, 2009.
[99] M. Ehrig and S. Staab, "QOM – quick ontology mapping," in *Proc. 3rd International Semantic Web Conference (ISWC)*, 2004, pp. 683–697.
[100] K. Saleem, Z. Bellahsene, and E. Hunt, "Porsche: Performance oriented schema mediation," *Information Systems*, vol. 33, no. 7-8, pp. 637–657, 2008.
[101] M. L. Lee, L. H. Yang, W. Hsu, and X. Yang, "XClust: clustering XML schemas for effective integration," in *Proc. 11th International Conference on Information and Knowledge Management (CIKM)*, 2002, pp. 292–299.
[102] F. Giunchiglia, A. Autayeu, and J. Pane, "S-match: an open source framework for matching lightweight ontologies," *Semantic Web Journal*, 2012, to appear.
[103] E. Jiménez-Ruiz and B. C. Grau, "Logmap: Logic-based and scalable ontology matching," in *Proc. 10th International Semantic Web Conference (ISWC)*, 2011, pp. 273–288.
[104] H.-H. Do and E. Rahm, "COMA – a system for flexible combination of schema matching approaches," in *Proc. 28th International Conference on Very Large Data Bases (VLDB)*, 2002, pp. 610–621.
[105] S. Duan, A. Fokoue, K. Srinivas, and B. Byrne, "A clustering-based approach to ontology alignment," in *Proc. 10th International Semantic Web Conference (ISWC)*, 2011, pp. 146–161.
[106] P. Lambrix and Q. Liu, "Using partial reference alignments to align ontologies," in *Proc. 6th European Semantic Web Conference (ESWC)*, 2009, pp. 188–202.
[107] Z. Aleksovski, "Using background knowledge in ontology matching," Ph.D. dissertation, Vrije U. Amsterdam, 2008.
[108] P. Jain, P. Hitzler, A. Sheth, K. Verma, and P. Yeh, "Ontology alignment for linked open data," in *Proc. 9th International Semantic Web Conference (ISWC)*, 2010, pp. 402–417.
[109] W. Hu, J. Chen, H. Zhang, and Y. Qu, "How matchable are four thousand ontologies on the semantic web," in *Proc. 8th Extended Semantic Web Conference (ESWC)*, 2011, pp. 290–304.
[110] R. Gligorov, W. ten Kate, Z. Aleksovski, and F. van Harmelen, "Using google distance to weight approximate ontology matches," in *Proc. 16th International World Wide Web Conference (WWW)*, 2007, pp. 767–776.
[111] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy, "Corpus-based schema matching," in *Proc. 21st International Conference on Data Engineering (ICDE)*, 2005, pp. 57–68.
[112] B. Saha, I. Stanoi, and K. L. Clarkson, "Schema covering: a step towards enabling reuse in information integration," in *Proc. 27th International Conference on Data Engineering (ICDE)*, 2010, pp. 285–296.
[113] V. Mascardi, A. Locoro, and P. Rosso, "Automatic ontology matching via upper ontologies: A systematic evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 609–623, 2010.
[114] P. Jain, P. Z. Yeh, K. Verma, R. G. Vasquez, M. Damova, P. Hitzler, and A. P. Sheth, "Contextual ontology alignment of LOD with an upper ontology: A case study with proton," in *Proc. 8th Extended Semantic Web Conference (ESWC)*, 2011, pp. 80–92.
[115] M. Sabou, M. d'Aquin, and E. Motta, "Exploring the semantic web as background knowledge for ontology matching," *Journal on Data Semantics*, vol. XI, pp. 156–190, 2008.
[116] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "Discovering missing background knowledge in ontology matching," in *Proc. 17th European Conference on Artificial Intelligence (ECAI)*, 2006, pp. 382–386.
[117] ——, "Semantic matching," *Encyclopedia of Database Systems*, pp. 2561–2566, 2009.
[118] ——, "Semantic schema matching," in *Proc. 13rd International Conference on Cooperative Information Systems (CoopIS)*, 2005, pp. 347–365.
[119] J. Shamdasani, T. Hauer, P. Bloodsworth, A. Branson, M. Odeh, and R. McClatchey, "Semantic matching using the UMLS," in *Proc. 6th European Semantic Web Conference (ESWC)*, 2009, pp. 203–217.
[120] M. Mochol, A. Jentzsch, and J. Euzenat, "Applying an analytic method for matching approach selection," in *Proc. 1st International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2006, pp. 37–48.
[121] M. Huza, M. Harzallah, and F. Trichet, "OntoMas: a tutoring system dedicated to ontology matching," in *Proc. 1st International Workshop on Ontology Matching (OM) at the International Semantic Web Conference (ISWC)*, 2006, pp. 228–323.
[122] M. Mochol and A. Jentzsch, "Towards a rule-based matcher selection," in *Proc. 16th International Conference on Knowledge Engineering: Practice and Patterns (EKAW)*, 2008, pp. 109–119.
[123] E. Peukert, J. Eberius, and E. Rahm, "Amc - a framework for modelling and comparing matching systems as matching processes," in *Proc. 27th International Conference on Data Engineering (ICDE)*, 2011, pp. 1304–1307.
[124] A. Algergawy, R. Nayak, N. Siegmund, V. Köppen, and G. Saake, "Combining schema and level-based matching for

web service discovery," in *Proc. 10th International Conference on Web Engineering (ICWE)*, 2010, pp. 114–128.
[125] K. Eckert, C. Meilicke, and H. Stuckenschmidt, "Improving ontology matching using meta-level learning," in *Proc. 6th European Semantic Web Conference (ESWC)*, 2009, pp. 158–172.
[126] A. Doan, P. Domingos, and A. Halevy, "Reconciling schemas of disparate data sources: A machine-learning approach," in *Proc. 20th International Conference on Management of Data (SIGMOD)*, 2001, pp. 509–520.
[127] V. Spiliopoulos and G. A. Vouros, "Synthesizing ontology alignment methods using the max-sum algorithm," *IEEE Transactions on Knowledge and Data Engineering*, 2011, PrePrints.
[128] C. Trojahn dos Santos, J. Euzenat, V. Tamma, and T. Payne, "Argumentation for reconciling agent ontologies," in *Semantic Agent Systems*, A. Eli, M. Kon, and M. Orgun, Eds. Springer, 2011, ch. 5, pp. 89–111.
[129] C. Domshlak, A. Gal, and H. Roitman, "Rank aggregation for automatic schema matching," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 4, pp. 538–553, 2007.
[130] F. Duchateau, Z. Bellahsene, and R. Coletta, "A flexible approach for planning schema matching algorithms," in *Proc. 16th International Conference on Cooperative Information Systems (CoopIS)*, 2008, pp. 249–264.
[131] H. Elmeleegy, M. Ouzzani, and A. K. Elmagarmid, "Usage-based schema matching," in *Proc. 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 20–29.
[132] A. Nandi and P. A. Bernstein, "Hamster: Using search clicklogs for schema and taxonomy matching," *PVLDB*, vol. 2, no. 1, pp. 181–192, 2009.
[133] N. Noy and M. Musen, "The PROMPT suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies*, vol. 59, no. 6, pp. 983–1024, 2003.
[134] F. Shi, J. Li, J. Tang, G. T. Xie, and H. Li, "Actively learning ontology matching via user interaction," in *Proc. 8th International Semantic Web Conference (ISWC)*, 2009, pp. 585–600.
[135] S. Duan, A. Fokoue, and K. Srinivas, "One size does not fit all: Customizing ontology alignment using user feedback," in *Proc. 9th International Semantic Web Conference (ISWC)*, 2010, pp. 177–192.
[136] S. M. Falconer and M.-A. D. Storey, "A cognitive support framework for ontology mapping," in *Proc. 6th International Semantic Web Conference (ISWC) and 2nd Asian Semantic Web Conference (ASWC)*, 2007, pp. 114–127.
[137] A. Mocan, E. Cimpian, and M. Kerrigan, "Formal model for ontology mapping creation," in *Proc. 5th International Semantic Web Conference (ISWC)*, 2006, pp. 459–472.
[138] A. Raffio, D. Braga, S. Ceri, P. Papotti, and M. A. Hernández, "Clip: a visual language for explicit schema mappings," in *Proc. 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 30–39.
[139] G. G. Robertson, M. P. Czerwinski, and J. E. Churchill, "Visualization of mappings between schemas," in *Proc. 12th Conference on Human Factors in Computing Systems (CHI)*, 2005, pp. 431–439.
[140] P. A. Bernstein and S. Melnik, "Model management 2.0: manipulating richer mappings," in *Proc. 26th International Conference on Management of Data (SIGMOD)*, 2007, pp. 1–12.
[141] E. von Hippel, *Democratizing Innovation*. MIT Press, 2005.
[142] P. Shvaiko, F. Giunchiglia, P. Pinheiro da Silva, and D. McGuinness, "Web explanations for semantic heterogeneity discovery," in *Proc. 2nd European Semantic Web Conference (ESWC)*, 2005, pp. 303–317.
[143] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos, "iMAP: Discovering complex semantic matches between database schemas," in *Proc. 23rd International Conference on Management of Data (SIGMOD)*, 2004, pp. 383–394.
[144] B. Alexe, L. Chiticariu, R. J. Miller, and W. C. Tan, "Muse: Mapping understanding and design by example," in *Proc. 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 10–19.
[145] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *Proc. 23rd International Conference on Data Engineering Workshops (ICDE Workshops)*, 2007, pp. 801–810.
[146] A. Zhdanova and P. Shvaiko, "Community-driven ontology matching," in *Proc. 3rd European Semantic Web Conference (ESWC)*, 2006, pp. 34–49.
[147] N. Noy, N. Griffith, and M. Musen, "Collecting community-based mappings in an ontology repository," in *Proc. 7th International Semantic Web Conference (ISWC)*, 2008, pp. 371–386.
[148] R. McCann, W. Shen, and A. Doan, "Matching schemas in online communities: A web 2.0 approach," in *Proc. 24th International Conference on Data Engineering (ICDE)*, 2008, pp. 110–119.
[149] N. Noy, A. Chugh, and H. Alani, "The CKC challenge: Exploring tools for collaborative knowledge construction," *IEEE Intelligent Systems*, vol. 23, no. 1, pp. 64–68, 2008.
[150] T. Tudorache, N. Noy, S. W. Tu, and M. A. Musen, "Supporting collaborative ontology development in Protégé," in *Proc. 7th International Semantic Web Conference (ISWC)*, 2008, pp. 17–32.
[151] S. Pinto, C. Tempich, and S. Staab, "Ontology engineering and evolution in a distributed world using DILIGENT," in *Handbook on Ontologies*, S. Staab and R. Studer, Eds. Springer, 2009, pp. 153–176.
[152] A. Doan, P. Bohannon, R. Ramakrishnan, X. Chai, P. DeRose, B. J. Gao, and W. Shen, "User-centric research challenges in community information management systems," *IEEE Data Engineering Bulletin*, vol. 30, no. 2, pp. 32–40, 2007.
[153] L. van der Meij, A. Isaac, and C. Zinn, "A web-based repository service for vocabularies and alignments in the cultural heritage domain," in *Proc. 7th Extended Semantic Web Conference (ESWC)*, 2010, pp. 394–409.
[154] J. Euzenat, "Alignment infrastructure for ontology mediation and other applications," in *Proc. International Workshop on Mediation in Semantic Web Services (MEDIATE)*, 2005, pp. 81–95.
[155] M. Ehrig, *Ontology alignment: bridging the semantic gap*. Springer, 2007.
[156] M. d'Aquin and H. Lewen, "Cupboard – a place to expose your ontologies to applications and the community," in *Proc. 6th European Semantic Web Conference (ESWC), Demo track*, 2009, pp. 913–918.

Pavel Shvaiko, PhD, is an innovation and research manager at Informatica Trentina SpA, Italy. He has provided various consulting services, including the eGovernment Interoperability Framework consultancy for the government of Mozambique, evaluation of research and/or innovation project proposals. He co-authored and co-edited a number of books, e.g., the first book published by Springer in the area of ontology matching together with Jérôme Euzenat, contributed to, and published in various international journals and conferences in the fields of Semantic Web, Artificial Intelligence, and Information Systems. He also ran a number of international initiatives as well as gave invited talks and tutorials devoted to the semantic integration themes. Specialties include: strategic consulting and planning, innovation management, research and business development with topics involving semantic heterogeneity and semantic integration.

Jérôme Euzenat is a senior research scientist at INRIA (Montbonnot, France). He has set up and leads the INRIA Exmo team dedicated to "Computer-mediated communication of structured knowledge" which is also part of the Laboratoire d'Informatique de Grenoble (Grenoble computer science lab). Dr Euzenat has contributed to reasoning maintenance systems, object-based knowledge representation, symbolic temporal granularity, collaborative knowledge base construction, multimedia document adaptation and semantic web technologies. His all time interests are tied to the relationships holding between various representations of the same situation. This covers connecting heterogeneous ontologies and interpreting their relations.